Hybrid Evolutionary Algorithm with Optimized Operators for Total Weighted Tardiness Problem

Yulia Zakharova

Sobolev Institute of Mathematics SB RAS

The research was supported by Russian Science Foundation grant N 22-71-10015

MOTOR–2023

Report Structure

- Problem statement
- Optimal recombination problem and its complexity
- Evolutionary algorithm with optimal recombination
- Computational experiment on OR-Library instances
- Conclusion and further research

Problem Statement $1|r_j, d_j|\sum_j w_j T_j$

Input data

n is the number of jobs;
$p_j \geq 0$ is the duration of job j;
$r_j \geq 0$ is the release date of job j;
$d_j \geq 0$ is the due date of job j;
$w_j \geq 0$ is the weight of job j.

Criterion

C_j is the completion time of job j;
$T_j = \max\{0; C_j - d_j\}$ is the tardiness of job j;
The goal is to construct a schedule such that the total weighted tardiness $\sum_{j \in \mathcal{J}} w_j T_j$ is minimized.

Evolutionary Algorithm with Steady State Replacement Scheme (EA)

1: Construct the initial population.
2: Until a termination condition is satisfied, perform steps
   2.1 Select two parent solutions π^1 and π^2.
   2.2 Apply mutation operator to parent solutions.
   2.3 Generate an offspring π' by applying a crossover operator to π^1 and π^2.
   2.4 Offspring π' replaces the worst individual of the population.
3: Perform local improvements of individuals of the last population.

Optimal Recombination Problem (ORP)

Statement

Given two parent solutions π^1 and π^2. It is required to find a permutation π' such that[a]:

(I) $\pi'_i = \pi^1_i$ or $\pi'_i = \pi^2_i$ for all $i = 1, \ldots, n$;

(II) π' has the minimum value of objective function $T(\pi')$ among all permutations that satisfy condition (I).

Optimal recombination may be considered as a best-improving move in a large neighbourhood defined by two parent solutions.

---

[a] N. Radcliffe, The algebra of genetic algorithms, Annals of Mathemathics and Artificial Intelligence 10 (4) (1994) 339–384.

Optimal Recombination in Evolutionary Algorithms

Balas E., Niehaus W. Optimized crossover-based genetic algorithms for the maximum cardinality and maximum weight clique problems (1998)

Yagiura M., Ibaraki T. The use of dynamic programming in genetic algorithms for permutation problems (1996)

Eremeev A.V. On complexity of optimal recombination for binary representations of solutions (2008)

Eremeev A., Kovalenko J. Optimal recombination in genetic algorithms for combinatorial optimization problems (2014)

Varadarajan S., Whitley D. The massively parallel mixing genetic algorithm for the traveling salesman problem (GECCO-19)

Eremeev A., Kovalenko Yu. A memetic algorithm with optimal recombination for the asymmetric travelling salesman problem (2020)

Chicano F., Ochoa G., Whitley D., Tinos R. Quasi-Optimal Recombination Operator (2019)

Restricted Even-Odd Partition (Du J., Leung Y.T.; 1990)

$A = \{a_1, a_2, \ldots, a_{2n_0}\}$:
$\sum_{a_i \in A} e_i = 2E,\ e_i > e_{i+1},\ i = 1, \ldots, 2n_0 - 1,$
$e_{2j} > e_{2j+1} + \delta,\ e_{2j-1} \leq e_{2j} + \delta,\ j = 1, \ldots, n_0,$
$\delta = \frac{1}{2} \sum_{i=1}^{n_0} (e_{2i-1} - e_{2i}).$

$$E = \frac{1}{2} \sum_{i=1}^{2n_0} e_i = \sum_{i=1}^{n_0} e_{2i-1} - \delta = \sum_{i=1}^{n_0} e_{2i} + \delta.$$

The question is to decide whether set A can be partitioned on two subsets A_1 and A_2 such, that:

$$\sum_{a_i \in A_1} e_i = \sum_{a_i \in A_2} e_i = E,\ |A_1| = |A_2| = n_0$$

and subset A_1 includes only one element from each pair $a_{2i-1},\ a_{2i}$.

NP-hardness

Theorem 1.

Optimal recombination problem (I)-(II) for $1|r_j = 0, d_j|\sum_j w_j T_j$ is NP-hard even in the case when all jobs except two have the same due dates.

Theorem 2.

Optimal recombination problem (I)-(II) for $1|r_j = 0, d_j|\sum_j T_j$ is NP-hard.

Theorem 3.

Optimal recombination problem (I)-(II) for $1|r_j, d_j|\sum_j T_j$ is NP-hard even in the case when we have only one job with individual release date and due date.

Solving Method for ORP

Main Idea

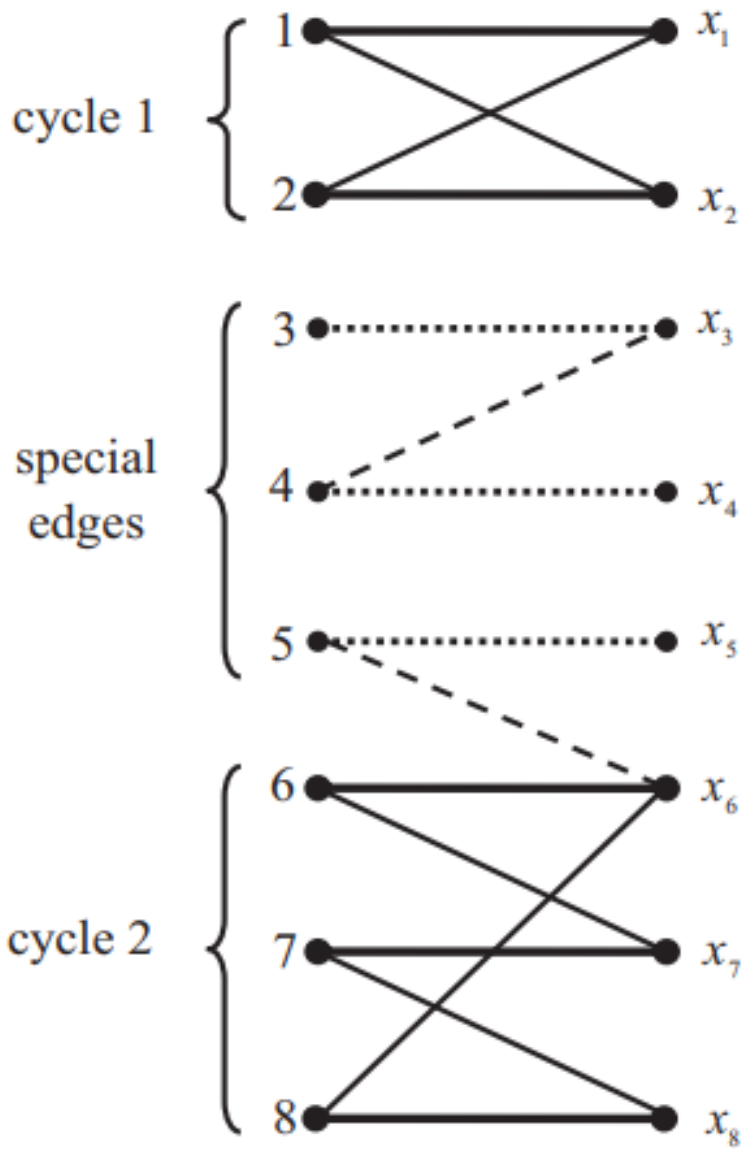
- $\bar{G} = (J_n, J, \bar{U})$ is the bipartite graph.
- $\bar{U} = \{\{i, j\} : i \in J_n,\ j = \pi_i^1 \ or \ j = \pi_i^2\}$ is the set of edges.
- Vertices of the left part $\leftrightarrow$ positions.
- Vertices of the right part $\leftrightarrow$ jobs.
- There is a one-to-one correspondence between the set of perfect matchings $\mathcal{W}$ in the graph $\bar{G}$ and the set Π of feasible permutations to a problem instance[a].

[a] Serdyukov A.I. (1978); Eremeev A., Kovalenko Yu. (2017)

Types of Edges

- An edge $\{i, j\} \in \bar{U}$ is called *special* if $\{i, j\}$ belongs to all perfect matchings in the graph $\bar{G}$.
- All edges, except for the special ones, are slit into cycles.

Example



Solving Method for ORP

Step 1. Build the bipartite graph $\bar{G}$, identify the set of special edges and cycles and find all maximal matchings in cycles.
Step 2. Enumerate all perfect matchings $W \in \mathcal{W}$ of $\bar{G}$ by combining the maximal matchings of cycles and joining them with special edges.
Step 3. Assign the corresponding solution $\pi \in \Pi$ to each $W \in \mathcal{W}$ and compute $T(\pi)$.
Step 4. Output the result $\pi^* \in \Pi$, such that $T(\pi^*) = \min\limits_{\pi \in \Pi} T(\pi)$.

Time Complexity

$O(n2^{q(\bar{G})})$, where $q(\bar{G}) \leq \lfloor \frac{n}{2} \rfloor$ is the number of cycles.

Solving ORP: Speed-up Procedure

Enumerating Solutions

Let permutation π is replaced by permutation $\pi' = (\pi_1, \ldots, \pi'_a, \pi'_{a+1}, \ldots, \pi'_b, \ldots, \pi_n)$ by changing the maximum matching in the cycle with the minimum index of the left part equal to a and the maximum index equal to b.

Computing Objective

The completion times of jobs in positions $a, a+1, \ldots, b$ are only changed for the permutation π' in comparison to the permutation π.

Almost All Feasible Instances

A graph $\bar{G} = (X_n, X, \bar{U})$ is called "good" if it satisfies the inequality $q(\bar{G}) \leq 1.1 \ln n$.

$\bar{\chi}_n$ is the set of "good" bipartite graphs $\bar{G} = (X_n, X, \bar{U})$.

χ_n is the set of all bipartite graphs $\bar{G} = (X_n, X, \bar{U})$.

$\frac{|\bar{\chi}_n|}{|\chi_n|} \to 1$ as $n \to \infty$ (Serdyukov A.I., 1978).

Theorem

Almost all pairs of parent permutations provide ORP instances having at most n feasible solutions and thus, they are solvable in $O(n^2)$ time.

Evolutionary Algorithm with Steady State Replacement Scheme (EA)

1: Construct the initial population.
2: Until a termination condition is satisfied, perform steps
   2.1 Select two parent solutions π^1 and π^2.
   2.2 Apply mutation operator to parent solutions.
   2.3 Generate an offspring π' by applying a crossover operator to π^1 and π^2.
   2.4 Offspring π' replaces the worst individual of the population.
3: Perform local improvements of individuals of the last population.

EA components

1. Initial population: greedy constructive heuristics and restrected local search.
2. Local optimization: insert and swap (Geiger M.J., Eur. Jour. Oper. Res. 2010).
3. Crossover: 0.8 **Optimized Cycle Crossover** (solving ORP); 0.2 **Order Crossover**.
4. Mutation: shift and swap.
5. Tournament selection.
6. The classic restarting rule.

Initial Population

Constructive heuristic

A non-scheduled job is selected at each step; the position is assigned to this job such that gives the best objective for the current partial solution.

Sequences of jobs

1. non-decreasing of due dates d_j
2. non-increasing of w_j/p_j
3. random
4. non-increasing of $(w_j/p_j) \cdot exp\{-\max\{0, (d_j - t - p_j)\}/(n * p_{aver})\}$

Previous Research

GA, Avci S. et al., IIE Transactions, 2003

A problem space algorithm for single machine weighted tardiness problems, **Pentium II 400MHz, 96Mb**

PVNS, Wang X., Tang L., Comp. Oper. Res., 2009.

A population-based variable neighborhood search for the single machine total weighted tardiness problem, **Pentium IV 3.0GHz, 512Mb**

m-VNS, Chung T.-P. et al.,Engineering Optimization, 2016

Multiple-variable neighborhood search for the single-machine total weighted tardiness problem, **Intel Core 2.3GHz, 2Gb**

HEA-DS, Ding J. et al., Comp. Indust. Eng., 2017

A hybrid evolutionary approach for the single machine total weighted tardiness problem, **Xeon E5440 2.83GHz, 16Gb**

OR-Library, $n = 40$

Algorithm	Time, sec.	n_{hit}	n_{opt}	Δ_{aver}
GA	29.11	–	100%	0
PVNS	6.19	100%	100%	0
m-VNS	0.908	100%	100%	–
HEA-OCX	0.05	100%	100%	0
HEA-DS	0.003	100%	100%	0

the number of instances is 125

50 runs on each instance

n_{hit} is the average percentage number of optimal solutions

n_{opt} is the percentage of instances, where the optimum is found

Δ_{aver} is the average relative deviation from the optimum

Intel Core i3-10100F CPU 3.60 GHz, 16 Gb

OR-Library, $n = 50$

Algorithm	Time, sec.	n_{hit}	n_{opt}	Δ_{aver}
GA	41.02	–	99.2%	0.000
PVNS	12.15	100%	100%	0
m-VNS	1.62	100%	100%	–
HEA-OCX	0.14	100%	100%	0
HEA-DS	0.006	100%	100%	0

the number of instances is 125

50 runs on each instance

n_{hit} is the average percentage number of optimal solutions

n_{opt} is the percentage of instances, where the optimum is found

Δ_{aver} is the average relative deviation from the optimum

Intel Core i3-10100F CPU 3.60 GHz, 16 Gb

OR-Library, $n = 100$

Algorithm	Time, sec.	n_{hit}	n_{opt}	Δ_{aver}
GA	118.97	–	66.4%	0.02
PVNS	183.47	100%	100%	0
m-VNS	5.845	99.84%	100%	–
HEA-OCX	1.5	100%	100%	0
HEA-DS	0.032	100%	100%	0

the number of instances is 125

50 runs on each instance

n_{hit} is the average percentage number of optimal solutions

n_{opt} is the percentage of instances, where the optimum is found

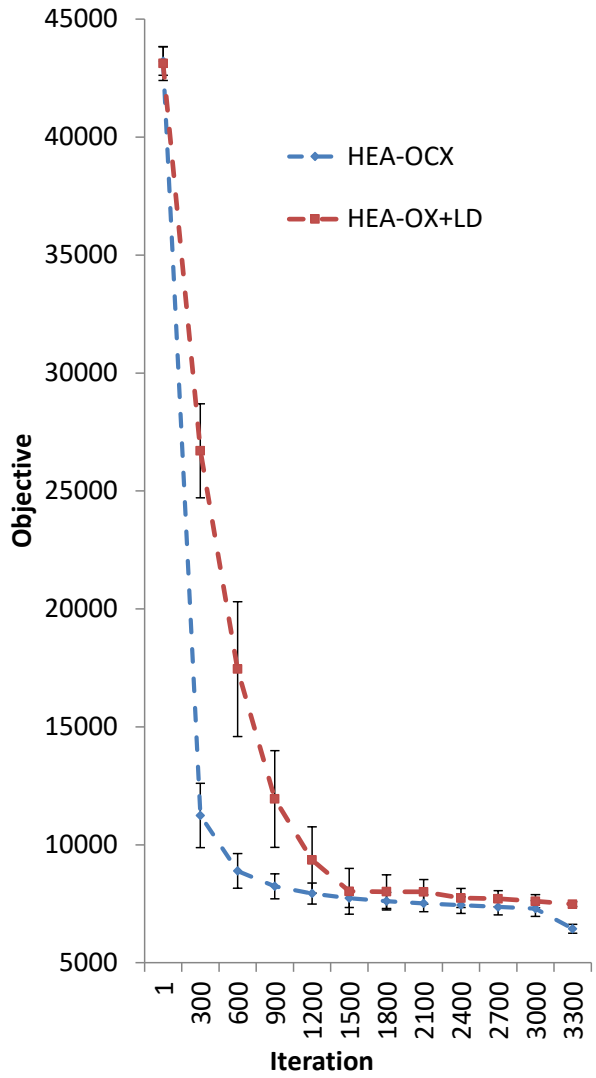
Δ_{aver} is the average relative deviation from the optimum

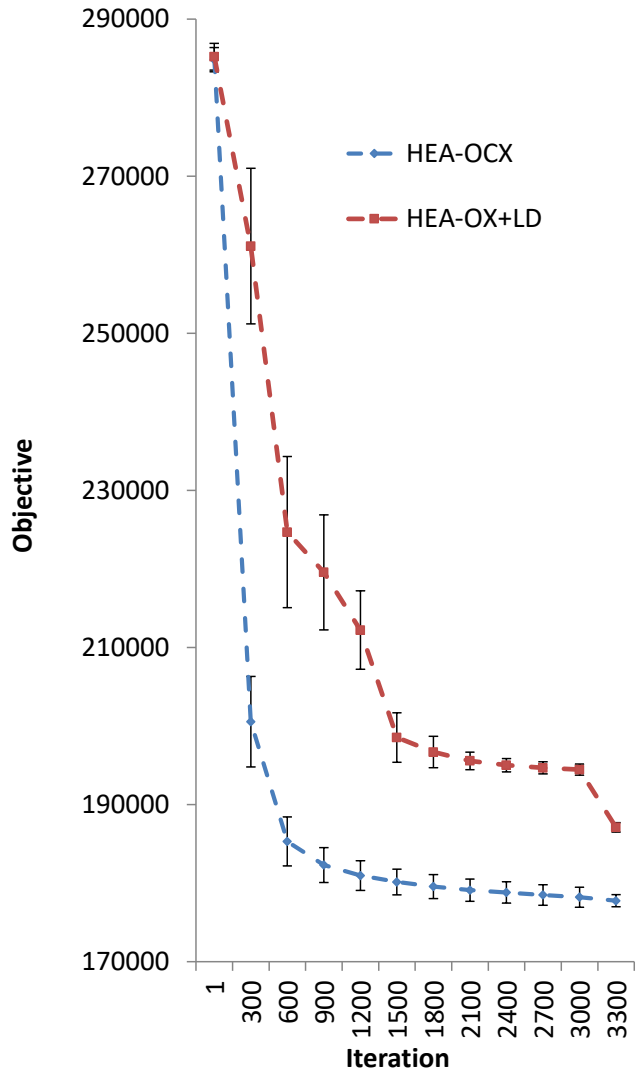
Intel Core i3-10100F CPU 3.60 GHz, 16 Gb

Other Operators

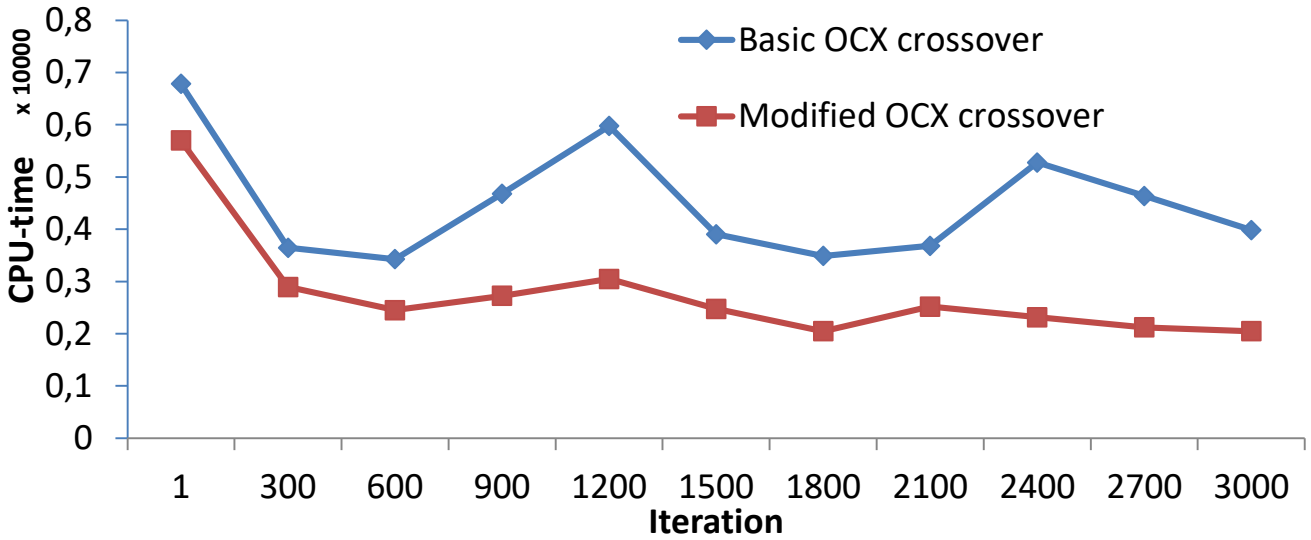
Operator	n_{bt}	n_{opt}	Δ_{aver}
	$n = 40$		
OCX	–	100%	0
CX	43/125	82.51%	0.0029
OX	20/125	93.73%	0.00078
OX+LD	17/125	94.56%	0.00076
	$n = 50$		
OCX	–	100%	0
CX	67/125	68.27%	0.0057
OX	41/125	84%	0.0017
OX+LD	38/125	92.88%	0.0013
	$n = 100$		
OCX	–	100%	0
CX	92/125	34.59%	0.0097
OX	90/125	40.56%	0.0055
OX+LD	80/125	62.32%	0.0031

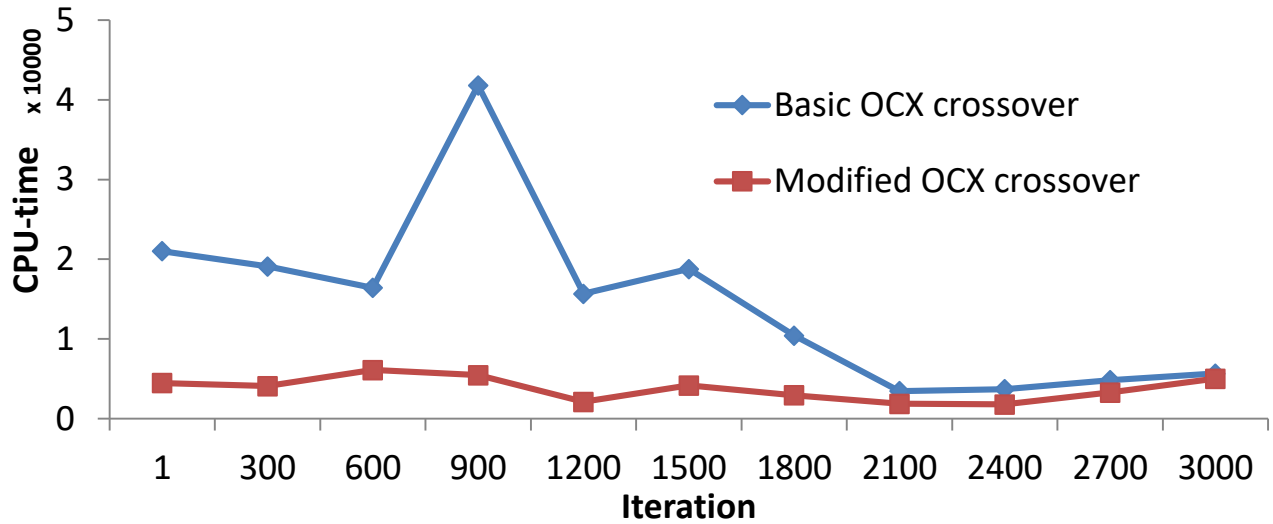
Dynamics of Objective Function





Dynamics of Utilized CPU Time (nanosec.)





Conclusion

- The NP-hardness of the Optimal recombination problem was proved for three practically important cases of $1|r_j, d_j|\sum_j w_j T_j$.
- A new speed-up procedure for solving the ORP was proposed based on the properties of criterion $\sum_j w_j T_j$.
- We developed new hybrid evolutionary algorithm with optimized operators for $1|r_j, d_j|\sum_j w_j T_j$.
- An experimental evaluation on OR-Library instances indicates that the proposed approach gives competitive results.

Thank you for your attention!